Sindhi Morphological Analysis: An Algorithm for Sindhi Word Segmentation into Morphemes

Waqar Ali Narejo, Javed Ahmed Mahar, Shahid Ali Mahar, Farhan Ali Surahio, Awais Khan Jumani
Department of Computer Science, Shah Abdul Latif University, Khairpur Mir's, Sindh, Pakistan

ABSTRACT-Morphological analysis is the process of constructing and deconstructing the words of a language, the process is based on the basic grammatical units which are stem, prefixes, suffixes and infixes. Sindhi is rich in morphological features with a great variety of affixes. The problem for Sindhi to come into computerization is the large number of variants in its morphology. This complexity is created due to different positions of prefixes, suffixes and stems in the words. The automatic word segmentation system normally faces such embedded hurdles in Sindhi language. An algorithm is required with a capability of dealing with such issues for the segmentation of Sindhi words. In this paper, an algorithm is designed and implemented to resolve the problem of segmenting Sindhi complex and compound words into possible morphemes. The developed words segmentation system has been tested on a list of 109 compound words, 179 prefix words, 1343 suffix words and 50 prefix-suffix words. The cumulative segmentation error rate of 5.02% is calculated. This system can also be used as pre-requisite in various Sindhi language and speech processing applications.

Keywords—Sindhi Morphology; Morphological Analysis; Word Segmentation; Morphemes

I. INTRODUCTION

Each natural language carries its specific and peculiar mechanism for generation of the words and conversion of other words from the root words. Morphology is a branch of linguistics which purely deals with the study of language from scientific point of view, concerning with words and their constructive grammatical units. The breaking or constructing units are prefix, infix, stem and suffix. Two types of words, i.e. basic and secondary are found in Sindhi. The basic words cannot be broken up any more but the secondary words are breakable and devisable into complex and compound words. The complex words are in the class of secondary words and are built by combining prefix/stem/suffixes. Compound words are formed with the combination of at least two words [1].

Sindhi, an Indo-Aryan language [2], bears a high degree of similarity with modern-day Urdu, Hindi and some other languages of northwest Indian sub-continent. There is also a firm relation among Sindhi, Arabic and Persian which is a contact-induced loaning and borrowing of many words from on to the other. The script used for Sindhi in South Asian states like Pakistan and India is purely Perso-Arabic on predominant basis. Apart from these regions, the same script is used by Sindhi migrants who are settled across the world. The Persian is involved in Arabic script for the representation of several letters which are not present in Arabic but are required to represent some implosive, retroflex and nasal sounds. The inflections and derivations in Sindhi script are most frequently found with the use of prefixes and suffixes. This proves the richness of Sindhi in terms of morphology. The issue arises due to a large number of morphological variants in Sindhi which is yet to be analyzed and solved successfully.

A compound word is usually formed by coalition of two or more simple words, i.e. گل گلاب (Rose), ڏينهن رات (Day Night). Prefix words which are the part of complex or derivative words are built with the union of prefix and stem or root words like گناه (a sin) is a primary word when combined with prefix بي (a prefix that shows opposite meaning) becomes بي گناه (innocent). In this, گناه (a sin) is a free morpheme and بي is the bound one [1]. Suffix words are the result of the combination of a root word and suffix, such as سمجھ (understanding) is a primary word when combined with suffix ڻ (a suffix that shows infinitive mood) becomes سمجھڻ (to understand). There are also many words in Sindhi dictionary which carry both the prefix and suffix along the stem or root word. The example of such words is ديس (Country) is a root word, in condition of its combination with prefix پر (a prefix that shows the sense of far), it forms پرديس (Abroad) and with addition of suffix ي, then it turns into پرديسي (Foreigner).

The automatic segmentation of words into morphemes through computer is what we call computational morphology. The morphological analysis is of assistance for many Natural Language Processing (NLP) applications working with large vocabularies [3]. For instance, it is traditional to preprocess texts by returning words to their original forms, specifically in text retrieval in morphologically enriched languages of the world. In computational applications, morphological analysis is basically the segmentation of words into tokens morphemes. The analysis separates the stem (core part of word) from the prefix (the letter-addition in the beginning of word) or the suffix (the letter-addition in the end of the word). Moreover, different approaches and methods have been proposed and developed for morphological deconstruction of words. They include statistical language modeling [4] [5], lexeme-based [6] [7], rule-based [8], syllable-based [9] and corpus-based [10].

II. LITERATURE REVIEW

For the past half-decade, many a great works have been published in the field of Sindhi linguistic applications, Rahman has worked Sindhi Morphology and Noun Inflections [1] in which he has discussed the variation of morphemes in nouns

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

with respect to the dialects used in Sindhi language. He has used addition, subtraction and replacement methods through which the basic morphemes are derived out in different forms due to the difference in the dialects. Apart from the computational perspective of the work, the grammatical discussion is also carried out like the numbers, genders and the cases of certain nouns. The conclusion of the research reveals that morphological construction of Sindhi language is either inflectional or derivational.

Sindhi is a rich language in terms of the characters having various glyphs. Such characters do also change their form within script depending on their position or order in the text. A Sindhi tokenization model is proposed by Mahar [11] having three layers, each layer assigned a separate task. Similarly, Bhatti [12] has worked on the Sindhi tokenization and developed a Sindhi word tokenization model. He has implemented several algorithms processing the tokenization of Sindhi text into individual words. This way, they have built a corpus and a word repository for grammar checking method, Sindhi Spellings and other NLP applications. The issue is dealt with the first encounter of sentence boundaries and extracting each sentence into a separate list form. In this list, each element is a complete sentence. The next step is the segmentation of sentences into words. This segmentation is performed on the basis of hard and soft spaces are taken as a part of word. Thus, the soft spaces are ignored of segmentation. The final step includes the filtration of words, removal of special characters, converting word into a token and saving it after the validation is done.

Sindhi is one of the Arabic script-based languages but its automatic segmentation application through morphological analyzer is yet unavailable. Though, Mahar [13] has developed four algorithms which possess the capability of segmenting the words into the root level, a higher degree of computational complexity regarding space and speed is the lapsing point of all of them. Due to its categorical function, Mahar's morphological analyzer uses the type of morpheme as its basis in each algorithm. Each algorithm works for a specific type of morpheme only so that process goes lengthy and slow for being an individualistic type. Therefore, a better and new algorithm is proposed for the segmentation of words into morphemes.

III. SINDHI MORPHOLOGY

Each language has its own grammar, foundation, and rules. Relatively, language is unique in its structure, function and application. For creating the awareness about morphology and analysis of words formation, some words are given in Table I. The first Sindhi word in Table I carries two morphemes: ann (اڻ) Jaan (جاڻ). The first morpheme is bound and the second is independent one. The slight change is notice in third word, the stem comes first and the addition at the second part. "پڑھ" is the root word whereas the added part ڻ is a suffix which is entailed to a word to change its meaning and sometimes word class even [14].

TABLE I: Comparative Morphological Analyses

Sindhi Morphology		
Word	**First Morphology**	**Second Morphology**
اڻجاڻ	اڻ	جاڻ
امر لهه	ا	م لهه
پ ڑهڻ	پ ڑھ	ڻ

A. Bound and Independent Morphemes

The bound morphemes are those smallest basic grammatical units which form their meaning when included in a word. Independently, they do not bear any meaning. Thus, the term suggests that they are bound with the words and do not stand independently having their meaning as a word. Consider the examples shown in Table II, the morphemes ڻ and و are the best examples of this type. Table III depicts the examples of Sindhi independent morphemes.

TABLE II: Bound Morphemes

Derivative	Root	Suffix/ Prefix
وڌهڻ	وڌھ	ڻ
پ ڑھ	پ ڑھ	و

TABLE III: Independent Morphemes

Word	Independent Morphemes	Stem
پرجوش	پ ر	وش ج
همخ يال	هر	خ يال

B. Zero Morphemes

There are several English words which are exactly identical in there different forms even. 'Sheep', 'fish', and 'deer' are some nouns which remain same in both plural and singular forms. Same is the case with some verbs like 'spread', 'shut' and 'put'. They remain same in their different forms of present and past. They are called homophonous. In both types of such words, whether nouns or verbs the phonological representation is zero. Therefore, these morphemes are known as zero morphemes.

In Sindhi, no such types of morphemes are found [14]. Though, we may find some homographic words in this regard which do not change their structure for changing into past or plural, they change their sound because in Sindhi, some words can make plural just by changing their diacritics with the same set of letters. The examples are shown in Table IV.

TABLE IV: Zero Morphemes

Singular	Plural
كَمُ	كَمَ
دَرُ	دَرَ
كِتابُ	كِتابَ

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

C. Root, Derivatives and Compound Words

Sindhi does also contain the same word types alike English: root words, derivatives and compound words. These types of words are depicted in the Tables V and VI. The first words in the Table 5 show the variation of meaning only with no change in word class of the root word. The changing of the meaning into the opposite of the root word defines the nature of prefix (بد), which is used to attach with the word for making its negation or opposition. The formation of second word suggests the uniqueness of Sindhi morphology in which the only letter (ا) is the suffix of the word. In addition to this, the letter (ا) used as suffix does not affect only the formation and meaning of the word but also changes its word class from verb to a noun. The prefixes of the third and fifth words are also of the same kind as of the first. They all mostly change the meaning of the word into its negative or opposite. The fourth word contains the suffix (ني) which is used for the emphasis only. It does not change the meaning or the class of the word.

TABLE V: Derivatives

Prefix/Suffix	Root Word	Complete Word
بد	بوءِ	بدبوءِ
ا	پوچ	پوچا
لا	شريڪ	لاشريڪ
ني	سپ	سپني
پر	ديس	پرديس

The first word in the Table VI contains two words as usual compound words do. The following words containing the same formation represent another property of such words which is the coalition of adjective and noun. Each of the words in the Table 6 is formed with one adjective and one noun. This endorses that most of the compound words in Sindhi possess the same nature in terms of their formation.

TABLE VI: Compound Words

Compound Word	First Word	Second Word
زهرپياڪ	زهر	پياڪ
ڏرتتي	ڏر	تتي
خوش بوءِ	خوش	بوءِ

IV. DATA COLLECTION

Corpus of language is inevitably essential for the computational exploitation. We have made the use of Sindhi corpus of 1, 05,733 words developed by Mahar [15], the sample of developed corpus is shown in Figure 1. It subsumes the genres of music, arts, politics, environment, and other texts. The sources for the collection of information were magazines, books of different types and newspapers. The data was collected in HTML and PDF formats. Then, they were converted into the fair equivalent formats of texts. Table VII represents the comprehensive details in figures for Sindhi corpus.

هاري جي احساس کي نٿا ڄاڻي سگهن جنهن جي گذر سفر جو واحد ذريعو اهي ئي به ايڪڙ هٿا
جيڪي پاڻي ڪوٽ جي نظر ٿي ويا انهي پاڻي جي آسري قرض کڻي پنهنجي ٻني ۾ هر ڪيڙيا ۽ بج
چٽيا تہ مٽان پاڻي جي ڪوٽ جي ڪري پاڻي جو وارو نہ اچي سگهيو ۽ سندن پوکيل بج سڏس ئي
اکين اڳيان سڪي تباہ ٿي ويو انهن اکين ۾ جيڪي حسرتون ۽ ارمان هوندا ڇا اهي ماڻهو انهن
ارمانن کي ڄاڻي سگهندا انهن حسرتن کي سمجهي سگهندا جيڪي هزارين ايڪڙ ٻنين جا مالڪ
هجن.

Fig.1. Sample of Developed Corpus

TABLE VII: Statistical Information of Sindhi Corpus

Corpus Type	Sentences	Word Tokens
Arts	1897	6884
Sports	1656	7582
Politics	2590	13351
Environment	1819	7098
Music	3822	15412
Total	**11,784**	**50,327**

A. Word Tokens

For the representation of the statistical information of this corpus, the first step was taken to break the text into the sentences. The second was the segmentation of sentences into words. This way the words were given to the system and it retained 50,327 unique word tokens. These word tokens do not represent the number of words in the given corpus but each word makes a token regardless of how many times it is used in the text.

Tokenization process is the segmentation of input objects of orthographic symbols into tokens [16]. This is the first prerequisite for NLP applications for these word tokens are then supplied to natural language processing applications for more computational processing. The word limits such as white space, digits, special signs and punctuation marks are useful for tokenization process. Apart from being useful, these sometimes also create complications in the process of tokenization. In this research, Mahar's tokenizer [17] is used which they proposed particularly for Sindhi language only. This model is composed of three layers which works consecutively one after the other as per the requirement. The implementation of the model, as done by Mahar, has also been imitated in this research work.

B. Developed Lexicon

A large lexicon is always required as a key component for the implementation of morphological analysis. It is, in general sense, a repository of words required to test the proposed algorithms. Hence, a lexicon for computational process is built with a collection of morphemes that are prefixes, suffixes and stems.

In print and electronic media, as the most of the Arabic script-based languages are written or typed without a variety of diacritic marks required for exactness of the sense, so is the case with Sindhi. Therefore, the basic limitation is the requirement of a fully diacritized corpus in order to build a lexicon. This may create another issue of the availability of

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

different versions of the same word with different diacritics in the lexicon. The words, then, may cause a great ambiguity with reference to their vocalization and meaning as well. Therefore, it is crucially essential to save all the words with full diacritics in the lexicon.

A lexicon having 50,327 words is built for the implementation of proposed algorithm. The lexicon is developed to segment Sindhi words into morpheme sequences. It has five tables and each table is used for the storage of separate type of word morphemes. The tables namely are root words, compound words, prefix words, suffix words and prefix-suffix words.

The developed lexicon is called Lexicon of Sindhi Morphological Analysis (LSMA). It is peculiarly constructed for proper and exact segmentation of words in Sindhi text. The lexicon contains only secondary type of words taken from the corpus. Table VIII represents the manifestation of secondary words.

TABLE VIII: Information of Secondary Words

Word Types	No. of Words
Compound	541
Prefix	893
Suffix	6713
Prefix-Suffix	247
Total	**8394**

V. SINDHI WORD SEGMENTATION ALGORITHM

A word is constructed with letters in a particular sequence. The letters first build a morpheme which is the smallest grammatical unit of language. Morphological segmentation is a general method for disintegration of a word into the combination of letters. This combination is a morpheme and cannot be further disintegrated. The development of any word segmentation technique requires one to be well aware of already developed and established techniques in order to bring effectiveness to the system.

A. Word Segmentation Technique

During the literature survey of Arabic morphological analysis techniques, it has been found that three morphological approaches are mostly in use, i.e. Table Lookup Approach, Combinatorial Approach and Linguistic Approach. These approaches can also be used for Sindhi word segmentation into its possible morphemes. Many times these approaches have been used for Arabic, Persian and Urdu languages. As Sindhi language belongs to the family of these languages on the basis of its script and nature so it can be predicted that these approaches can stand useful for Sindhi.

In this paper, Table Lookup Approach is used for the segmentation of Sindhi words into possible morphemes. This approach mainly relies on a considerably large set of tables in which Sindhi words are stored and found in natural texts with their morphemes. Morphemes are set in the forms of stem, suffix and prefix. A variety of words are found in a language, i.e. foreign words, functional words and proper nouns which require a unique place in the table. Multiple entries may also be found with the same structure which is due to the fact of different types of sense relations of words among them. The sense relations include homonymy, metonymy, synonymy, hyponymy and synonyms and antonyms. Few of these relations require a word to be spelt same but meant differently. These entries enable the system to be capable of dealing with multiple analyses of the words.

The entries in these tables are stored in alphabetical letter. For the optimization of search through vertical and horizontal order, a hash table stands efficient and effective to be used. In addition to this, a compression or precision technique is also possible to be used effectively for the reduction of storage needs. Thus, it makes the morphological analysis quite simple by accessing hash table.

B. Proposed Sindhi Word Segmentation Algorithm

The lexicon driven approach is used for our proposed algorithm, therefore, a lexicon named LSMA is constructed that stores all possible morphemes, and the lexicon consists on five tables {T1, T2, T3, T4, T5}.

The database table T1 is constructed for storing all the possible root words. The database table T2 is constructed for storing the compound words with three column vectors T2= {C1, C2, C3}. The column C1 is used to store the complete compound word, C2 is used for storing first word and C3 is used for storing second word.

In lexicon LSMA, database table T3 is constructed for storing words having prefix morpheme, it has three column vectors T3={C1, C2, C3} where, C1 is used for storing prefix along with primary word, C2 is used for storing prefix morphemes and C3 is used for storing the primary word.

The database table T4 is used for storing words having suffixes. It has three column vectors T4= {C1, C2, C3}. Each column is responsible to store the segments of words after its breakage. Thus, C1 is used for storing words having suffix along with primary word, C2 is used for storing suffix morphemes and C3 is used to store primary word.

The database table T5 is used for storing the words having both prefix and suffix morphemes at a time, this table consists of five column vectors T5= {C1, C2, C3, C4, C5}, where C1 is used for storing the complete words having prefix and suffix morphemes, C2 is used for storing only prefix and C3 is used for storing only suffix morphemes, C4 is used for storing primary word and C5 is used for storing the primary word along with suffix morpheme. Prefix and suffix lexicon entries cover all possible concatenations of Sindhi prefixes and suffixes.

Algorithm of Sindhi Word Segmentation

1. Input Sindhi Text
2. Tokenize Input Text
3. Store all word tokens into temporary array WORDTEMP
4. Select words one by one from WORDTEMP
5. Search Selected word from Column 1 of Table T1 //To check that it is a root word or not
6. If search is successful then display message “This is a Root Word” and go to step 16

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

7. Else split selected word into characters and store them into temporary array CHTEMP
8. Search selected word from Column 1 of Table T2 // **For Compound Words**
9. If Search is successful then
    a. Repeat until either both words are successfully compared or any word is not found in Table T2
        i. Select characters consecutively from CHTEMP and append into VAR1
        ii. Search and compare VAR1 from Column 2 of Table T2
        iii. If search is successful then
            1. Concatenate remaining characters of CHTEMP and store into VAR2
        iv. Else go to Sub-step a
        v. Search and Compare VAR2 from Column 3 of Table T2
        vi. If search is successful then
            1. Display "First Word", VAR1 and "Second Word", VAR2
    b. End
10. Else search selected word from Column 1 of Table T3 //**For Prefix Words**
11. If search is successful then
    a. Repeat until both conditions are true or any morpheme is not found in Table T3
        i. Select characters consecutively from CHTEMP and append into VAR1
        ii. Search and Compare VAR1 from Column 2 of Table T3
        iii. If search is successful then
            1. Concatenate remaining characters of CHTEMP and store into VAR2
        iv. Else go to Sub-step a
        v. Search and compare VAR2 from Column 3 of Table T3
        vi. If search is successful then
            1. Display "Prefix", VAR1 and "Root Word", VAR2
    b. End
12. Else search selected word from Column 1 of Table T4 //**For Suffix Words**
13. If Search is successful then
    a. Repeat until both conditions are true or any morpheme is not found in Table T4
        i. Select characters consecutively from CHTEMP and append into VAR1
        ii. Search and compare VAR1 from Column 3 of Table T4
        iii. If search is successful then
            1. Concatenate remaining characters of CHTEMP and store into VAR2
        iv. Else go to Sub-step 1
        v. Search and compare VAR2 from Column 2 of Table T4
        vi. If search is successful then
            1. Display "Root Word", VAR1 and "Suffix", VAR2
    b. End
14. Else search selected word from Column 1 of Table T5 //**For Prefix-Suffix Words**
15. If search is successful then
    a. Repeat until all conditions are satisfied or any morpheme is not found in Table T5
        i. Select characters consecutively from CHTEMP and append into VAR1
        ii. Search and compare VAR1 from Column 2 of Table T5
        iii. If search is successful then
            1. Concatenate remaining characters of CHTEMP and store into VAR2
        iv. Else go to Sub-step a
        v. Search and compare VAR2 from Column 4 of Table T5
        vi. If search is successful then
            1. Display "Prefix", VAR1
        vii. Split VAR2 into characters and store into array SUTEMP
        viii. Select characters consecutively from SUTEMP and append into SVAR1
        ix. Search and Compare SVAR1 from Column 4 of Table T5
        x. If search is successful then
            1. Concatenate remaining characters of SUTEMP and store into SVAR2
        xi. Else go to Sub-step viii
        xii. Search and compare SVAR2 from Column 3 of Table T5
        xiii. If search is successful then
            1. Display "Root Word", SVAR1 and "Suffix", SVAR2
    b. End
16. End

The process of proposed algorithm starts with the input step of the text. The text can be input through two ways; it can be typed and produced to the system and can also be taken from the corpus of the language. Once the text is input, the process has begun. The input text is tokenized at the beginning of the process. The tokenization model of Mahar [18] has been used in this system. The tokenization sends the prepared tokens to an array called WordTemp. This array stored the word tokens so that they can be forwarded forth. The system then takes each token from WordTemp one by one and starts searching the match for the selected word. The first search is carried out in Table1 Column1. If the search is successful, system displays the word as a "Root Word". The process does not go further for the search is over and the match is found. This is because

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

we have stored the root words in Table1 Column1 and the successful search witnesses the word as a root one. If the search is unsuccessful and match is not found, the control shifts to the next search step. Before moving to the next search, the system splits the word into separate characters that constitute it and stores them in an array called CHTEMP.

VI. IMPLEMENTATION AND RESULTS

After the details for the familiarization of our developed algorithm, the algorithm is taken into its application in the system. The application process is defined in this section along with the results received after the application. The results are not calculated at a whole but for the acute evaluation of the system, we have categorized the process into different parts. The system has been evaluated through separate classes of words i.e. prefix words, suffix words, prefix-suffix word and compound words.

The performance of algorithm is evaluated by rating the correctly and incorrectly segmented words as given in [19]. Moreover, the segmentation error rate with each word class is calculated so that the vivid and transparent results can be obtained. These separate word class results will also help find the causes and issues that reduce the success rate of the system. This calculation standard is used under the influence of [19], Segmentation error rate (SER) is defined as:

(Number of incorrectly segmented words/ total number of word) x 100

A. Compound Words

The main algorithm first makes it sure that the word is not a root one then it shifts to the search of the forwarded word in Table2 Column1 for searching if the word if compound one. The successful search shifts control of main algorithm to the Module Compound Words. The process begins by taking in the split letters stored in CHTEMP one by one until VAR1 is formed by achieving a match from Column2 of this table. Once, the match is found and VAR1 is formed, the system generates VAR2 taking the remaining letters from CHTEMP. The forming of VAR1 requires a repetition process by appending letters one by one from CHTEMP. VAR2 is formed and it also requires a condition of must-match in Column3 of the table. When both conditions are fulfilled and VAR1 and VAR2 are formed the system displays the result by showing first word as VAR1 and second word as VAR2.

After the selection of a word, the splitting into separate characters takes place and each character is selected one by one and all these characters are being appended and stored into a temporary generated variable. Then, system compares the contents of this variable with T2-C2. If the characters are matched properly, the concatenation of remaining characters starts and then these remaining characters and fed into another variable and again the comparison starts with T2-C3, in case of successful match, the system displays both words. For example, ڈینهن رات (Day Night), each character is taken into process from right to left like ر and it is compared with column C2, then ا is selected and both are appended together and again compared with C2. The system consecutively selects third character ت and again all are appended and compared with C2. The successful search leads to the concatenation of remaining characters ڈ, ي, ن, ه, ن through the same procedure and comparison takes place with B3. After both conditions are fulfilled, words are displayed as word1 ڈینهن and word2 is رات.

In order to scrutinize and verify the system efficiency and performance, we took 109 words randomly for testing. These words were taken from training dataset of 541 words. The number of taken words stands 20% of training data. For experimental purpose, compound words were categorized into two classes; the words having a hard space in between like گل گلاب (Rose) and the words having no hard space like پلپل (Every Moment). The gist of results is given in Table IX. The pictorial representation of word SER is given in Figure 2.

TABLE IX: Segmentation Error Rate using Module COMPOUND

Compound Word Classes	No. of Words	Correct	Incorrect	SER
With Hard Space	82	82	0	0.0
Without Hard Space	27	26	1	3.7
Total	**109**	**108**	**1**	**3.7**

The complication of compound words is observed during the process of morphological analyzer. It is particularly observed with the words having connecting letters in between the compound words. This leads to the erroneous depiction of morphemes in such situations. In addition to this, certain compound words have non-connective letters in between. They lead to another erroneously segmented morpheme for the remaining non-connecting letters in the second word form a word that has an entirely different meaning from the actual sense of the whole compound word. Thus, two erroneous morphemes are segmented by analyzer in this case. The situation leads to an increase in SER of the morphological analyzer. Due to these issues, the SER of the proposed morphological reached 3.7% with the compound words having no hard space in between and 0.0 with those having hard space.

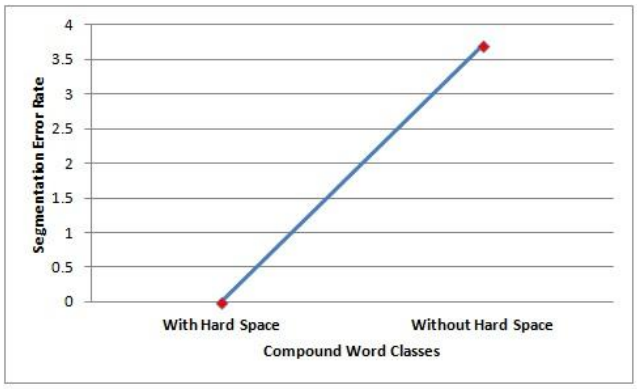

Fig.2. Word SER using Module COMPOUND

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

B. Prefix Words

The second step of search ends in two conditions; the word is compound and process shifts by executing Module Compound Word and second condition is taking the process to a search in Table3 Column1. The successful search in this table starts the execution of Module Prefix Words. The PREFIX module receives a word from main algorithm as input and splits it into characters, system selects each character one by one and appends it into temporary generated variable and then compares the contents of this variable with T3→C2, if comparison is successful, then concatenates remaining characters and stores into another variable and compares it with T3→C3, if search is successful, then system displays both morphemes. For example, بيوفا (unloyal), the system selects each character from right to left like ب and compares it with column C2, then selects character ي and appends it as next character and compares with C2, if search is successful, then concatenates remaining characters و, ف, ا and compares with C3, when both conditions are satisfied, then system displays prefix بي and root وفا.

The appending of letters and searching for a match in Column3 is repeatedly performed till VAR1 is formed and match is sought out in Column3. The VAR2 is formed by appending the remaining letters together and the search is performed in Column3 of the table. Column3 has the root words in it. It is also understood that formulation of VAR1 extracts the prefix from the word and leaves the remaining letters which must form a stem and VAR2 as well. VAR1 is compared with the words stored in Column2 and VAR2 is compared with the words stored in Column3. After achieving both matches, the system shows the result as VAR1 "Prefix" and VAR2 "Root Word".

Evaluating the performance of this module, 179 words were randomly taken from the training dataset containing 893 words. The words having prefixes are classified into three categories: (1) The prefix words showing the sense of negation like بد بخت (unlucky) (2) The prefix words showing the sense of adjective like لاجواب (matchless) and (3) The prefix words showing the sense of antonym پرديس (abroad). The summary of results is shown in Table X. The SER of negation, adjective, and antonym is depicted in Figure 3.

TABLE X: Summary of Results using Module PREFIX

Prefixes Classes	No. of Words	Correct	Incorrect	SER
Negation	68	67	1	1.47
Adjective	97	94	3	3.09
Antonym	14	14	0	0.0
Total	**179**	**306**	**10**	**4.56**

The calculated results depict that the SER of Negation and adjective is higher than that of Antonym. Since the prefixes used to form a negative or opposite sense to that of the original meaning of the particular word can stand as a word separately with their own meaning. Such prefixes are also used as in individual word in Sindhi text. Therefore, morphological analyzer segments them as a separate word sometimes and its SER increases relatively. On the other hand, simple Antonyms having prefixes are segmented successfully with the SER of 2.3% which is lesser than that of Negation and Adjectives.

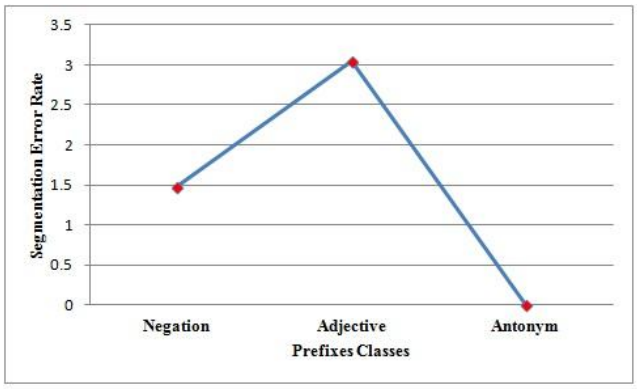

Fig.3. Word SER using Module PREFIX

C. Suffix Words

The SUFFIX module takes the word given as input and split it into characters, system selects each character one by one and appends it into temporary generated variable, and then, compares the contents from T4→C3, if comparison is successful, then concatenates remaining characters and compares it with T4→C2, if search is successful, then system displays both morphemes. For example, سيني (All to all), the system selects each character from right to left like س and compares it with column C3, then selects character پ and appends it as next character and compares with C3, if search is successful, then concatenate remaining characters ن, ي and compares it with C2, when both conditions are satisfied, then system displays root word سپ and suffix ني.

The process of this module begins with the input of separately stored letters of the selected word in CHTEMP. One by one, the letters are brought in till VAR1 is formed. After the formulation of VARI the module searches for its match in Column2. Column2 is responsible to store the root words therefore VARI in this module is the formulation of root words. The appending of letters and searching their match in Column is repeatedly done till its formulation and final match in Column2. After VAR1, the module appends all the remaining letters and forms VAR2 which is a suffix and such type is stored in Column3. VAR2 is compared with the combination of letters stored in Column2 to find its match. After achieving the successful matches of VAR1 and VAR2 in their respective columns, the system displays result as VAR1 "Root Word" and VAR2 "Suffix".

The number of words taken randomly for testing from the training dataset was 1343. The total number of words in training dataset was 6713. The selected sample was taken in order to gauge the performance of this module. For experimental purpose, words with suffixes were categorized into 5 classes: (1) the suffix words in singular sense like بکيو

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

(Hungry) (2) the suffix words of plurality like سونارا (Jewlars) (3) the suffix words showing adjectival meaning like پاڳيريو (Lucky) (4) the suffix words classed in masculine like چوڪرو (Boy) and (5) the suffix words of feminine like گهرواري (Wife). The summary of results with the standard of SER is given in Table XI. The graphical representation of results is given in Figure 4.

TABLE XI: Summary of Results using Module SUFIX

Suffixes Classes	No. of Words	Correct	Incorrect	SER
Adjective	631	622	9	1.43
Singular	112	210	2	1.79
Plural	102	99	3	2.94
Masculine	181	179	2	1.10
Feminine	317	312	5	1.58
Total	**1343**	**5,566**	**47**	**8.84**

The depiction of results proves Masculine class to be yielding the least SER in all. On the other hand Feminine class as well as Singular has acceptable level results with 1.79% and 1.85% SERs respectively. The cumulative SER is 8.84%. This is due to the Plural class of suffix words which stands with an SER of 2.94%. Due to this class, the performance of whole system is affected and led to a higher level of SER. The reduction of SER in plural will ultimately improve the performance of the system. Eventually, besides Singular and Plural, the results are considerably better and encouraging as well.

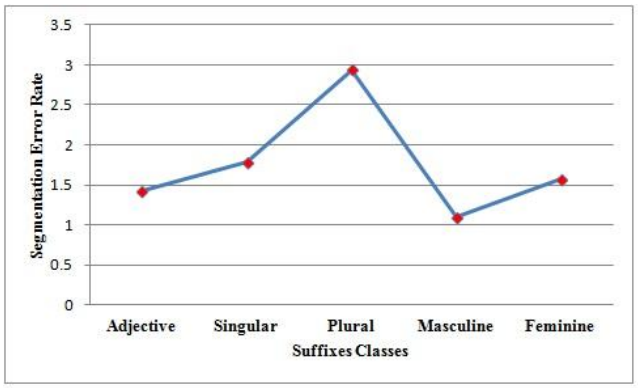

Fig.4. Calculated SER using Module SUFIX

D. Prefix-Suffix Words

The process of Module Suffix Words begins with the input of separately stored letters of the selected word in CHTEMP. One by one, the letters are brought in till VAR1 is formed. After the formulation of VARI the module searches for its match in Column2. Column2 is responsible to store the prefix morphemes therefore VARI in this module is the formulation of prefix morphemes. The appending of letters and searching their match in Column2 is repeatedly done till its formulation and final match in Column2 is found. After VAR1, the module appends all the remaining letters and forms VAR2 which is the remaining part of the word containing the root and suffix and such type is stored in Column4. VAR2 is compared with the combination of letters stored in Column4 to find its match.

After achieving the successful matches of VAR2 in its respective column, the system displays result as VAR1 "Prefix". After producing the result of VAR1 Prefix, the system concatenates the VAR2 and split it into letters. The split form is stored in another array called SUTEMP. The selection of letters one by one from SUTEMP and appending them again starts till a SVAR1 is formed. After forming SVAR1, the system starts searching the match for SVAR1 from Colum4 where primary words are stored. If system succeeds to find the match, it concatenates the remaining letters taken from SUTEMP and forms SVAR2. Then SVAR2 is compared with the words stored in Column3.

After finding the match of SVAR2 in Column3, the system displays the result as SVAR1 "Root Word" and SVAR2 "Suffix". It is noted that the concatenation and appending of the letters from TEMPs are repeatedly done till the search comes successful. The "Else" condition drives the system to jump to the previous step of concatenation and appending of letter and continues it till the match is found in the column.

This module is based on two phases: system segments prefix and the stem in first phase and it cuts off suffix from the root word in the second. The word is appointed into the module from the main algorithm as input and concatenates it into separate characters. System takes each character one by one respectively and keeps appending them into a temporarily generated variable. While appending the characters it also keeps on comparing the contents of this variable from T5→C2. As the comparison comes to a successful match, then the remaining letters are concatenated and stored into another variable. Once more, the splitting and appending takes place and storing the characters into variable while comparing them with T5→C4. Till the match comes successful during comparison process, then the rest of the letters are stored and the process repeats itself again undergoing each step that are already described. After the successful match while comparing the contents with T5→C3, the system displays three parts of the word.

For example, پ ردي سي (Foreigner), the system takes each character from right to left i.e. پ and compares it the contents in C2, it selects ر and appends to the previous character and again compares with C2, after successful search it concatenates the rest of the letters, د ي, ي, س and takes them through the same process. When the stem دي س is successfully segmented, it looks for the other characters ي and does comparison with the contents of C3, after the fulfillment of all three conditions; system shows a display of prefix پ ر root word دي س and suffix ي. A list containing 247 words was prepared for training 50 words. These words were tested through the system in this module. The outcomes are shown in Table XII.

TABLE XII: Results using Module PREFIX-SUFFIX

No. of Words	Correct	Incorrect	SER
50	46	4	8.0

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

E. Cumulative Results

The developed morphological analyzer has been gauged in testing 109 compound words, 179 prefix words, 1343 suffix words and 50 prefix-suffix words. The overall results showed the SER of 5.02%. The calculated cumulative word segmentation error rate of different word classes is given in Table XIII. The Figure 5 depicts the cumulative segmentation error rate of the system in graphical form.

TABLE XIII Cumulative SER of Each Word Types

Types of Words	Segmentation Error Rate
Compound	3.7
Prefix	4.56
Suffix	8.84
Prefix-Suffix	8.0
Cumulative SER	**5.02**

The results show that compound words have resulted the least SER which is encouraging part of the work. The SER of these words is 3.7% cumulatively. Segmentation of suffix words produces an SER of 8.84% and the reason of its height is already described as the Adjectives with suffixes sometimes stand as completely separate words in Sindhi script. The SERs produced after suffix words and prefix-suffix words are at a little difference of 0.84%.

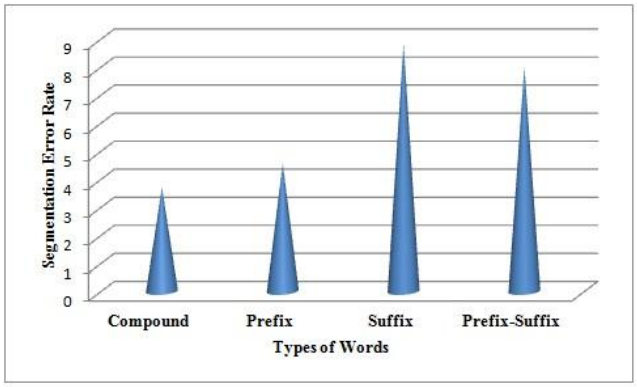

Fig.5. Cumulative Words SER of Proposed Algorithm

VII. WORD SEGMENTATION APPLICATION

In spite of all the details given about the developed algorithm, its function, application and results, the need of more clarity remains intact for the understanding the whole research and its processed outcome through the system. The interface contains two boxes that are connected with the process of the given text. The upper box is responsible to show the text that is input into the system. This box not only accommodates the direct typing of the text but has a property of receiving an already developed file as its input. The system processes the text that is directly typed. In otherwise case, it receives the files which are in doc. format only.

After the text is input into the system, the user has to click the Process Menu and a pop-up will appear in a drop-down box. The box has three options i.e. Apply, Data Setting and Clear. As the user will click the Apply button, this will activate the system to take the text for processing. The process ends up by showing the results in the output box of the interface. The depiction of input box and outbox are totally different in terms of the organization of the text. The input box takes the plain text as it is typed. The input and output box depicts the results in six different columns as shown in Figure 6. These columns have been assigned their respective morphemes. Each word from the text is processed and put into its respective column. The columns are given the names of the morphemes found in Sindhi language. Each column receives a particular morpheme taken out of the word after segmentation.

Fig.6. Text Input and Output Interface

VIII. DISCUSSION AND CONCLUSION

Sindhi language has been considered as one the most complex languages when it comes to automatic language applications. The abundance of homographs in orthography of Sindhi and its cursiveness prove the above fact. Such nature embeds the hurdles in the way of word segmentation process. The affixes become rather complex to be segmented due to cursiveness. Thus, the developed word segmentation system is designed in such a way so that it should segment these basic grammatical units as an embarking source to NLP Applications. The proposed algorithm possesses the capability to deal with all the basic grammatical units of Sindhi: Root words, Prefixes and Suffixes and provides the base for segments its words into morphemes.

The two data sets are extracted from our developed lexicon for experimental purpose: the training data set and the testing data set. The testing data set contains 109 compound words, 179 prefix words, 1343 suffix words and 50 suffix-prefix words. After the process of words segmentation, compound words yielded the SER of 3.7%, the prefix words gave an SER of 4.56%, and suffix words did 8.84% and prefix-suffix words 8%. The individual calculation and cumulative segmentation error rates of the proposed algorithm derive out that the results have come up to the acceptable level. Although, 5.02% SER is produced, the correct segmentation supports the effectiveness of the proposed algorithm with an exactitude rate of 94.08%. It is a proven fact that Sindhi word segmentation is an essential for its application in any natural language processing task.

The received results have achieved an acceptable level, though; they are not up to the mark as they should be. This

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

piece of research has paved a way to reach the ultimate accuracy in NLP applications for Sindhi language. The current SER is surely possible to decrease in future; the achieved SER is a little high due to the limited lexicon. The SER can easily be decreased if the lexicon is extended to a great extent. The table lookup approach is used for automatic word segmentation system. If the approached as combined at least two, the algorithm will be more useful for the same task. Hence, in future, we shall also test the combined system of combinatorial and linguistic approaches.

REFERENCES

[1] Rahman, M. U., "Sindhi Morphology and Noun Inflections", Proceedings of the Conference on Language & Technology, Lahore-Pakistan, pp. 74-81, 2009.

[2] Bakhash, S. W., "Sindhi Boli Jo Sarf Ain Nahuo", Sindhi Adabi Board, Jamshoro, 2006.

[3] Creutz, M., Lagus, K., "Unsupervised Discovery of Morphemes", Work shop on Morphological and Phonological Learning of ACL, Philadelphia, Pennsylvania, USA, pp. 21-30, 2002.

[4] Lee, Y. S., Papineni, K., Roukos, S., "Language Model Based Arabic Word Segmentation", In the 41st Annual Meeting of the Association for Computational Linguistics, Sappora, Japan, pp. 399-406, 2003.

[5] Vergyri, D., Kirchhoff, K., Duh, K., and Stolcke, A., "Morphology-Based Language Modeling for Arabic Speech Recognition", proceedings of the International Conference on Spoken Languages, Volume3, Jeju, Korea, pp. 2245–2248, 2004.

[6] Buckwalter, T., "Buckwalter Arabic Morphological Analyzer Version 1.0", Linguistic Data Consortium, Catalog Number LDC2002L49, ISBN 1-58563-257-0, 2002.

[7] Habash, N., "Large Scale Lexeme Based Arabic Morphological Generation", Traitement Au-tomatique du Langage Naturel, Fez, Morocco, pp. 271-276, 2004.

[8] Constantine, L., Erwin, C., Mitchell, P. Marcus, Charles, Y., "A Rule-Based Unsupervised Morphology Learning Framework", In Working Notes of the 10th Workshop of the Cross-Language Evaluation Forum, Corfu, Greece, 2009.

[9] Cahill, L., "A Syllable-based Account of Arabic Morphology", In Abdelhadi Soudi, Antal van der Bosch and Günther Neumann (eds.) Arabic Computational Morphology Dordrecht: Springer, pp. 45-66, 2007.

[10] Itai, A., Segal, E., "A Corpus Based Morphological Analyzer for Unvocalized Modern Hebrew", the Workshop on Machine Translation for Semitic Languages: Issues and Approaches, 9th Machine Translation Summit, New Orleans, pp. 29-36, 2003.

[11] Mahar, J. A., Shaikh, H., Memon, G. Q., "A Model for Sindhi Text Segmentation into Word Tokens", Sindh University Research Journal (Science Series), Vol. 44, No. 1, pp. 43-48, March 2012.

[12] Bhatti, Z., Ismaili, I. A., Soomro, W. J., Hakro, D. N., "Word Segmentation Model for Sindhi Text", American Journal of Computing Research Repository, Vol. 2, No. 1, pp. 1-7, 2014.

[13] Mahar, J. A., Memon, G. Q., Danwar, S. H., "Algorithms for Sindhi Word Segmentation using Lexicon Driven Approach", International Journal of Academic Research, Vol. 3, No. 3, pp. 28-35, May 2011.

[14] Narejo, W. A., Mahar, J. A., "Morphology: Sindhi Morphological Analysis for Natural Language Processing Applications", IEEE International Conference on Computing, Electronic and Electrical Engineering, Quetta, Pakistan, 2016.

[15] Mahar, S. A., "Comparative Analysis of Vowel Restoration for Arabic Script Based Languages Using N-Gram Models", MS Thesis, Department of Computer Science, Shah Abdul Latif University, Khairpur Mir's, pp. 31-32, 2014.

[16] Attia, M. A., "Arabic Tokenization System", In the Proceedings of the Workshop on Important Unresolved Matters, Prague, Czech Republic, pp.65-72, 2007.

[17] Nguyen, T., Vogel, S., "Context-based Arabic Morphological Analysis for Machine Translation", Proceedings of the 12th Conference on Computational Natural Language Learning, pp. 135–142, 2008.

[18] Shah A. A.; Ansari, A. W.; Das, L., "Bi-Lingual Text to Speech Synthesis System for Urdu and Sindhi", National Conference on Emerging Technology, pp. 126-130, 2004.

[19] Lee, Y. S., Papineni, K., Roukos, S., "Language Model Based Arabic Word Segmentation", the 41st Annual Meeting of Association for Computational Linguistics, Sappora, Japan, pp. 399-406, 2003.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500